

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе 2
по дисциплине «Алгоритмы и структуры данных»
Тема: Рекурсия

Студент гр. 8304

Матросов Д.В.

Преподаватель

Фирсов К.В.

Санкт-Петербург

2019

Цель работы.

Освоение рекурсивного метода решения задач.

Задание.

Вариант 27.

Реализовать рекурсивную функцию вычисления детерминанта квадратной матрицы по формуле:

$$\det A = \sum_{i=1}^n (-1)^{i+1} a_{i1} \bar{M}_j^1$$

Где \bar{M}_j^1 - дополнительный минор к элементу a_{i1} матрицы A.

Дополнительный минор элемента a_{i1} матрицы A n-го порядка есть определитель порядка n-1, соответствующий той матрице, которая получается из матрицы A путём вычёркивания i-й строки и j-го столбца.

Описание алгоритма работы.

Программа использует следующий алгоритм для выполнения задачи: функция считающая детерминант получает на вход исходную матрицу, если размерность матрицы не превышает 2, то в результирующую переменную записывается определитель матрицы, считающийся следующим образом:

Если размерность матрицы равен 2m, то

$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = a \times d - c \times b$$

Если размерность равна 1, то

$$\det(a) = a$$

В том случае, если размерность исходной матрицы превышает 2, то функция входит в цикл по первой строке матрицы, на каждой итерации цикла функция вызывает себя для соответствующего минора. Функция поступает подобным образом до тех пор, пока не будет возможно посчитать `det` способами приведенными выше, для размерности 2 и 1.

Описание функций и структур.

1. Написана функция `int main()`, отвечающая за открытие необходимых для ввода файла, ввод исходных данных, общение с пользователем и запуск функции-обработчика.

2. Основная функция `double det(vector<vector<double>> matrix)`, принимающая на вход исходную матрицу, представленную в виде вектора векторов, для удобства поиска минора. Возвращает посчитанное значение детерминанта матрицы `matrix`. Эта функция либо считает детерминант матрица размерностью < 3 либо вызывает себя рекурсивно n -раз, передавая в качестве параметра вызов функции `minor(matrix, i)`, где $0 \leq i < n, n = \text{размерности матрицы}$.

3. Функция `vector<vector<double>> minor(vector<vector<double>> matrix, int number)`, выполняющая нахождения минора матрицы `matrix`, путем удаления 1-й строки и столбца с номером `number`. Возвращает минор в виде вектора векторов.

Тестирование.

Ввод	Вывод
<code>matrix_size: 2</code> <code>matrix:</code> <code>1 1</code> <code>1 1</code>	<code>DetA = 0</code>
<code>matrix_size: 1</code> <code>matrix:</code> <code>3.5</code>	<code>DetA = 3.5</code>

matrix_size: 2 matrix: 1 1 -1 1	DetA = 2
matrix_size: 4 matrix: 2 4 1 1 0 2 0 0 2 1 1 3 4 0 2 3	DetA = 0
matrix_size: 1 matrix: 2 1 12 1 0 1 2 0 0 0 13 2 0 0 0 1	DetA = 26
matrix_size: 1 matrix: 0.5 1 2 1 0.5 2 2 2 0.5	DetA = 3,625
matrix_size: 1 matrix: 0.5 1 -2 1 -0.5 2 -2 2 0.5	DetA = -8,625

Выводы.

В ходе выполнения лабораторной работы была написана рекурсивная функция вычисления детерминанта матрицы, как было предложено в задании.

ПРИЛОЖЕНИЕ А. КОД ПРОГРАММЫ.

```
#include <stdio.h>
#include <tchar.h>
#include <fstream>
#include <iostream>
#include <string>
#include <vector>

using namespace std;

vector<vector<double>> minor(vector<vector<double>> matrix, int
number) {

    int matrix_size = matrix.size();

    vector<vector<double>> minor;

    for (int i = 1; i < matrix_size; i++) {
        minor.push_back(matrix[i]);
        minor[i-1].erase(minor[i-1].begin() + number);
    }
    return minor;
}

double det(vector<vector<double>> matrix) {
    double result = 0;
    int matrix_size = matrix.size();
    if (matrix_size == 1) {
        result = matrix[0][0];
    }
    else if (matrix_size == 2) {
```

```

        result = matrix[0][0] * matrix[1][1] - matrix[1][0] *
matrix[0][1];
    }
    else {
        for (int i = 0; i < matrix_size; i++) {
            result += matrix[0][i] * pow(-1, i + 2) *
det(minor(matrix, i)); //+2 т.к. i >= 0
        }
    }
    return result;
}

int main()
{
    cout << "Enter full name of input file: ";

    string input_file_name; //имя файла ввода
    cin >> input_file_name;
    ifstream fin;
    fin.open(input_file_name);

    vector<vector<double>> matrix; //исходная матрица

    vector<double> line;

    if (fin.is_open()) {
        double tmp;

        int matrix_size; //размерность матрицы

        fin >> matrix_size;

```

```

        for (int i = 0; i < matrix_size; i++) {
            for (int j = 0; j < matrix_size; j++) {
                fin >> tmp;
                line.push_back(tmp);
            }
            matrix.push_back(line);
            line.clear();
        }

        cout << "DetA = " << det(matrix) << endl;

    }
    else {
        cout << "File " << input_file_name << " can't be opened
for reading!";
    }

    fin.close();
    fout.close();
    system("pause");
    return 0;
}

```